The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 18 February 2003

Patents Form 7/77

The Patent Office

7/77

Patents Act 1977
Rule 15

Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920020047GB1
2.	Patent application number *(if you know it)*	0229724.0
3.	Full name of the or of each applicant	INTERNATIONAL BUSINESS MACHINES CORPORATION
4.	Title of invention	APPARATUS, METHOD AND COMPUTER PROGRAM FOR DEFINING A DATA MAPPING BETWEEN TWO OR MORE DATA STRUCTURES
5.	State how the applicant(s) derived the right from the inventor(s) to be granted a patent	By employment and by agreement
6.	How many, if any, additional Patents Forms 7/77 are attached to this form?	
7.		I/We believe that the person(s) named over the page (and on any extra copies of this form) is/are the inventor(s) of the invention which the above patent application relates to.

Signature
R J Burt

18 December 2002
Date

8.	Name and daytime telephone number of person to contact in the United Kingdom	N Watson Tel: 01962 818955

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Graham Castree CHARTERS
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number *(if known)*

Patents ADP number *(if known)*

If there are more than three inventors, please write their names and addresses on the back of another Patents Form 7/77 and attach it to this form

Patents ADP number *(if known)*

REMINDER

Have you signed the form?

Patents Form 1/77

THE PATENT OFFICE
K
19 DEC 2002
RULE 97
NEWPORT

Patents Act 1977
Rule 16

The Patent Office

1/77

23DEC02 E772671-1 D00611
P01/7700 0.00-0229724.0

Request for grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920020047GB1
2.	Patent application number *(The Patent Office will fill in this part)*	0229724.0
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	INTERNATIONAL BUSINESS MACHINES CORPORATION Armonk New York 10504 United States of America
	Patents ADP number *(if you know it)*	5196370001
	If the applicant is a corporate body, give the country/state of its incorporation	State of New York United States of America
4.	Title of the invention	APPARATUS, METHOD AND COMPUTER PROGRAM FOR DEFINING A DATA MAPPING BETWEEN TWO OR MORE DATA STRUCTURES
5.	Name of your agent *(if you have one)*	R. J. Burt
	"Address for Service" in the United Kingdom to which all correspondance should be sent *(including the postcode)*	IBM United Kingdom Ltd Intellectual Property Dept. Hursley Park Winchester Hampshire SO21 2JN
	Patents ADP number *(if you know it)*	7903925001

6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	Country	Priority App No *(if you know it)*	Date of filing *(day/month/year)*
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date or the earlier application	No of earlier application		Date of filing *(day/month/year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.)

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	8
Claim(s)	4
Abstract	1
Drawing(s)	6 + 6

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	2
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application

Signature
R J Burt

18 December 2002
Date

12. Name and daytime telephone number of person to contact in the United Kingdom

N Watson
01962 818955

APPARATUS, METHOD AND COMPUTER PROGRAM FOR DEFINING A DATA MAPPING BETWEEN TWO OR MORE DATA STRUCTURES

Field of the Invention

The invention relates to the field of data transformations, and more specifically to the definition of such transformations.

Background of the Invention

Distributed systems typically comprise a multitude of heterogeneous applications all communicating using different languages. In order for two such different applications to communicate with one another, it is necessary that data in a format A from the first application is transformed into data in a format B understood by the second application. Figure 1a shows a first example of the components that enable such a transformation to take place.

Application 10, by way of example, uses a SAP internal data format. In order to communicate with application 50, a request from application 10 may go via a message broker/intermediary system 30. Adapter 20 interfaces with Application 10 and transfers the SAP internal formatted message to broker 30. At the broker it is determined that the message is destined for application 50 which uses an Ariba internal format. The broker therefore transforms the message received from application 10 into an Ariba internal message format suitable for transferring the message to application 50. Upon receipt of this message, adapter 40 interfaces with application 50 and communicates the Ariba formatted message.

It should however be appreciated from the above that the number of individual transformations required can be huge. A formula for determining the number of transformations is n*n-1, where n is the number of data types used (e.g. message sets, where a message set is the set of messages understood by one application), and we are defining transformations in both directions.

For this reason an alternative solution was developed. Referring to figure 1b, a "standard" format for communication is agreed upon by adapters 20 and 40. One example of such a format is the Business Object Document (BOD) specification defined by the Open Applications Group. When application 10 wishes to communicate with application 50, adapter 20 converts the data into BOD form which is received by adapter 40 and transformed into the Ariba data format. The number of transformations now is 2*n. Therefore for small numbers of applications there is no benefit,

(e.g. 2 applications = 4 transformations vs. 2 in the original design of figure 1a), but for larger numbers of applications the benefits are important (e.g. 5 applications = 10 transformations vs. 20 in the original design of figure 1a).

(Note, figures 1a and 1b show different integration topologies however this is not relevant to the transformation reduction. It is possible, for example, to achieve the same results by transforming to the "standard" format in the intermediary system.)

Nevertheless, it will be appreciated that a highly labour intensive activity when performing Enterprise Application Integration is the definition of data/message transformations. Each message set can be large and complex and typically consists of a number of different messages each containing a variety of different fields. (For example the OAG BOD standard version 7.1 has over 180 different messages.) Ordinarily the user selects source and target messages and a tool presents them side by side. The user then defines the relationships between fields in the source message and fields in the target message. With reference to figure 2 it can be seen that message set A has a "part" message containing the fields "name"; "id"; "price"; and "description". Message set B has a corresponding message and fields but uses different terms to refer to these. Thus a user has to identify that the "part" message in message set A corresponds to the "item" message in message set B. The user then has to map the fields within the "part" message to the fields within the "item" message. Thus "name" is mapped to "prodname"; and "ID" is mapped to "identifier" etc. Note, this example is simple in that there is only one message in each set and there is a one to one correspondence between the fields. The reality is however typically far more complicated in that there may be numerous message sets; messages and fields to contend with and that there is not necessarily a one to one correspondence between the fields in two messages. Thus it is typically an onerous task to define the required transformations between messages in different message sets.

Summary of the Invention

Accordingly the invention provides, in a first aspect, an apparatus for defining a data mapping between two or more data structures comprising: two or more data structure comprising incompatible identifiers; storage for storing said two or more data structures; means for selecting said two or more data structures; and means for deriving a definition of a data mapping between data elements represented by said incompatible identifiers, wherein said means for deriving a data mapping definition is operable to analyse previous data mapping definition information.

Preferably the previous data mapping definition information comprises user defined information.

Preferably the user is provided with a plurality of possible data mapping definitions. These can be prioritised to the user based on at least one predefined rule. Such prioritisation makes the task of selecting a mapping from the plurality of possibilities easier. A variety of different rules for the prioritisation process are preferably possible (e.g. a previous user selection).

Preferably the two or more data structures are grouped into sets, a first data structure of the two or more data structures forming part of a first set and a second data structure of said two or more data structures forming part of a second set. Preferably previous data mapping definition information comprises at least one of:

i) a previous data mapping definition between two data structures, one from the first set and one from the second set;

ii) a previous data mapping definition between two data structures, one from the first or second set and the other from another set; and

Iii) a previous data mapping definition between two data structure which do not come from the first or second set.

Such information is preferably used in the prioritisation process. For example from a plurality of possible data mappings i) may be ranked more highly than ii) and ii) may be ranked more highly than iii).

In the preferred embodiment the mapping definition information concerns messages of message sets. Preferably the information comprises at least one of:

i) a message field to message field definition; and

ii) a message name to message name definition.

Preferably it is possible to use reverse mapping definition information for defining a data mapping. Figure 4 provides an example of this where StaffNumber.TimeServed has previously been mapped to Employee.YrsServ. Thus this information is used, in the example, to map Employee.YrsServ to PersonnelNumber.TimeServed.

Note the apparatus may be located at an intermediary system such as a message broker.

The invention is preferably implemented in software.

According to another aspect, the invention provides a method for defining a data mapping between two or more data structures comprising the steps of: selecting said two or more data structures; and deriving a definition of a data mapping between data elements represented by said incompatible identifiers, wherein said deriving step comprises analysing previous data mapping definition information.

Brief Description of the Drawings

A preferred embodiment of the present invention will now be described by way of example only and with reference to the following drawings:

Figures 1a and 1b illustrate an overview of enterprise application integration (which includes message transformation) according to the prior art;

Figure 2 illustrates a defined correspondence between two message sets according to the prior art; and

Figures 3a, 3b, 4 and 5 illustrate message transformation according to a preferred embodiment of the present invention.

Detailed Description

With reference to figures 3a and 3b, two message sets (MS) are selected by a user (A and B, step 100; 105). A source message and a target message are then selected by the user (step 110). (In this example the source message is Part and the target message is Item.) From message Part a field (Name) is chosen (step 120). It is determined whether there is any previous transformation definition information which might be of use here (step 130) and since there isn't the user defines this transformation, mapping the Name field to ProdName in the Item message of message set B (step 140). Information regarding this transformation is held in non-volatile storage for possible future use (step 140). (Note, there may not always be a corresponding field to map to in a target message - see below.) Following the same process, the user also defines Part.ID and Part.Price. As can be seen from figure 3b, these are mapped to Item.Identifier and Item.Price (steps 160; 120; 130; 140). There is no

corresponding field for Part.Description in the Item message and so the transformation for this field is not defined.

Having defined transformations for all the fields in the Part message for which there are corresponding fields in the Item message, it is determined at step 170 that there is another source message (Order) in set A and a target message in set B between which transformations are to be defined (step 110). Field ID is selected from this message (step 120). Part.ID was previously defined as mapping to Item.Identifer, thus it is deduced that any field named ID in message set A is likely to map to any field named Identifier in message set B (step 130). In message set B a PurchaseOrder message exists and this message includes the field Idenfier. Thus a suggestion is made to the user that Order.ID might map to PurchaseOrder.Identifer. The user chooses to accept PurchaseOrder.Identifier as the correct definition of Order.ID and thus this recommendation is actioned and information regarding this choice is added to non-volatile memory (step 155; 165). The next field in message Order is Quantity (steps 160; 120). Quantity is not a field that has been seen before and so the user defines its correspondence to PurchaseOrder.Quantity and information regarding this is added to non-volatile memory (steps 130, 140). However with Order.Price, the system has previously seen that Part.Price maps to Item.Price and therefore suggests that Order.Price might map to PurchaseOrder.Price (steps 160; 120; 130; 150). The user then chooses to accept this recommendation and it is actioned and information regarding this choice added to non-volatile memory (step 155, 165) The process continues with StockCheck.ID (steps 160; 170; 110; 120). Previously Part.ID was mapped to Item.Identifer; and Order.ID was mapped to PurchaseOrder.Identifer. The system thus deduces that StockCheck.ID might well map to StockLevel.Identifier (steps 130; 150). In this example, the user chooses to accept the recommendation and this is actioned and information regarding this action is stored in non-volatile memory (steps 155, 165). Finally StockCheck.Quantity possibly maps to StockLevel.Quantity based on the previous transformation of Order.Quantity to PurchaseOrder.Quantity (steps 160; 120; 130; 150). Again this is accepted and actioned (step 155, 165). Since there are now no more messages in set A (step 170), it is determined whether there are any more message sets for which transformation are to be defined (step 180). (Note this may mean defining a transformation between a current message set and a new message set or between two completely new message sets.) If there are any more message sets, then the process returns to step 105 and starts over again. Otherwise, the process ends at step 190.

The system can aid the user in a number of different ways. Prioritisation of recommendations is discussed in more detail later;

however it will be briefly discussed here. For example, if the user has defined Order.ID as mapping to PurchaseOrder.Identifier, thus it is known to the system that there is a correspondence between the Order message in set A and the PurchaseOrder message in set B. It can use this information to prioritise suggestions about possible future transformation definitions (e.g. Order in message set A might map to PurchaseOrder in previously unseen message set C). Further the storage of information at step 165 can be used to prioritise suggestions. For example, the previous definition information used to make the current recommendation may have come from a transformation between two different messages sets (see below), if the user selects that recommendation for messages sets A and B this information can be stored to prioritise this recommendation for other transformation definitions relating to the same two message sets (A & B).

It will now be appreciated by one skilled in the art that the flow described above relates to just one way in which the invention could be implemented. For example, in an alternative embodiment, the tool first analyses all the messages in two message sets and makes a series of recommendations. The user can then address recommendations for each field in turn, choosing to accept or reject these. Any fields for which there are no recommendations, or for which the user does not like the suggested recommendations, are left to the user to define.

It will no doubt also now be appreciated by one skilled in the art that transformations for all messages in a message set may not be required. Further, a one to one mapping has been shown here. In practice n messages may be mapped to m messages (for example three messages may map to two messages.)

The suggestions for possible transformation definitions do not have to come from the same message set. Figure 4 shows message sets C, D, E, F and G. Sets C and D relate to personnel records and the correspondence between messages (one shown) in the two sets have been defined prior to defining mappings for message sets E and F. Message sets E and F relate to catering records. The fact that Name in the employee message of set C is defined as mapping to FullName in the PersonnelNumber message of set D is used to suggest to the user that Employee.Name in message set E may map to PersonnelNumber.FullName in message set F. Further if the transformations between messages in set C and D are being defined, information from previous transformation definitions involving another set and C or D can be used. In the example, StaffNumber.TimeServed (message set G) has been mapped to Employee.YrsServ (message set C). This information can be used to suggest that Employee.YrsServ may map to PersonnelNumber.TimeServed in message set D. (This assumes that the previously defined mapping works in

reverse.) Correspondence between message names as well as message fields may also be used. For example, the fact that the user has defined a link between the Employee message in set C and the PersonnelNumber message in set D may be used to suggest a link between the Employee message in set E and the PersonnelNumber message in set F. Such information is useful in prioritising suggestions to the user regarding field definitions.

When defining transformations between two message sets C and D, suggestions could be prioritised to the user based on some predefined rules. For example the priorities could be as follows:

1. Information from existing C and D message set transformation definitions has top priority.

2. Information from transformation definitions including one of message set C or D is prioritised next (e.g. C and G)

3. Information from any other transformation definition is prioritised last (e.g. E and F)

A tool implementing the invention is preferably implemented in computer software. This tool could be provided with the message broker/intermediary system, or adapter software (e.g. as shown in figures 1a and 1b. The components of such a tool according to a preferred embodiment are shown in figure 5.

The tool 200 comprises a selection component 210. Using this component, the user can select two message sets between which to define transformations. Having made this selection, an analyser 220 component is invoked which scans messages in the selected message sets. For each message and field, within the message sets, the analyser determines whether it knows of previous transformation information which might be useful with regard to the defining each message and field transformation. In order to do this, analyser component 220 consults previous transformation definition information held in non-volatile storage 230. If it finds helpful information within storage 230, it uses such information to suggest possible definitions to the user via suggestion component 240. The user can then use selection component 210 to choose one of the suggested definitions. If on the other hand no such useful information is held within storage 230, user definitions component 250 enables the user to define the correspondence between a message/field in the source message set and a message/field in the selected destination message set. This definition is then stored in storage 230 for possible future use.

In this way the previously onerous task of defining transformation information is alleviated.

It will be appreciated that whilst the invention has been defined in terms of messages and messaging systems, the invention is not limited to such and is applicable to any environment where data of one format needs to be converted to data of another format.

Note, throughout the specification the terms transformation and mapping are used interchangeably.

CLAIMS

1. Apparatus for defining a data mapping between two or more data structures comprising:

two or more data structure comprising incompatible identifiers;

storage for storing said two or more data structures;

means for selecting said two or more data structures; and

means for deriving a definition of a data mapping between data elements represented by said incompatible identifiers,

wherein said means for deriving a data mapping definition is operable to analyse previous data mapping definition information.

2. The apparatus of claim 1 wherein the previous data mapping definition information comprises user defined information.

3. The apparatus of claim 1 or 2 wherein the deriving means comprises means for providing a user with a plurality of possible data mapping definitions.

4. The apparatus of claim 3, comprising:

means for prioritising the plurality of possible data mapping definitions based on at least one predefined rule.

5. The apparatus of claim 3 or 4 comprising:

means for selecting one of said plurality of possible data mapping definitions.

6. The apparatus of claim 5 comprising means for using a user selection in prioritising the plurality of possible data mapping definitions.

7. The apparatus of any preceding claim, wherein said two or more data structures are grouped into sets, a first data structure of said two or more data structures forming part of a first set and a second data structure of said two or more data structures forming part of a second set, and wherein previous data mapping definition information comprises at least one of:

i) a previous data mapping definition between two data structures, one from the first set and one from the second set;

ii) a previous data mapping definition between two data structures, one from the first or second set and the other from another set; and

iii) a previous data mapping definition between two data structure which do not come from the first or second set.

8. The apparatus of claim 7, wherein from a plurality of possible data mappings, a previous data mapping definition between two data structures, one from the first set and one from the second set, is ranked more highly than a previous data mapping definition between two data structures, one from the first or second set, and the other from another set.

9. The apparatus of claim 8, wherein from a plurality of possible data mapping definitions, a previous data mapping definition between two data structures, one from the first or second set and the other from another set is ranked more highly than a previous data mapping definition between two data structures which do not come from the first or second set.

10. The apparatus of any preceding claim, wherein data mapping definition information concerns messages of message sets.

11. The apparatus of claim 10 wherein previous data mapping definition information comprises at least one of:

i) a message field to message field definition; and

ii) a message name to message name definition.

12. The apparatus of any preceding claim comprising:

means for using reverse mapping definition information for defining a data mapping.

13. A method for defining a data mapping between two or more data structures comprising the steps of:

selecting said two or more data structures; and

deriving a definition of a data mapping between data elements represented by said incompatible identifiers,

wherein said deriving step comprises analysing previous data mapping definition information.

14. The method of claim 13 wherein the previous data mapping definition information comprises user defined information.

15. The method of claim 13 or 14 wherein the deriving step comprises providing a user with a plurality of possible data mapping definitions.

16. The method of claim 15 comprising the step of:

prioritising the plurality of possible data mapping definitions based on at least one predefined rule.

17. The method of claim 15 or 16 comprising the step of:

selecting one of said plurality of possible data mapping definitions.

18. The method of claim 17 comprising the step of using a user selection in prioritising the plurality of possible data mapping definitions.

19. The method of any of claims 13 to 18, wherein said two or more data structures are grouped into sets, a first data structure of said two or more data structures forming part of a first set and a second data structure of said two or more data structures forming part of a second set, and wherein previous data mapping definition information comprises at least one of:

i) a previous data mapping definition between two data structures, one from the first set and one from the second set;

ii) a previous data mapping definition between two data structures, one from the first or second set and the other from another set; and

iii) a previous data mapping definition between two data structure which do not come from the first or second set.

20. The method of claim 19, wherein from a plurality of possible data mappings, a previous data mapping definition between two data structures, one from the first set and one from the second set, is ranked more highly than a previous data mapping definition between two data structures, one from the first or second set, and the other from another set.

21. The method of claim 20, wherein from a plurality of possible data mapping definitions, a previous data mapping definition between two data structures, one from the first or second set and the other from another set is ranked more highly than a previous data mapping definition between two data structures which do not come from the first or second set.

22. The method of any of claims 13 to 21, wherein data mapping definition information concerns messages of message sets.

23. The method of claim 22 wherein previous data mapping definition information comprises at least one of:

i) a message field to message field definition; and

ii) a message name to message name definition.

24. The method of any of claims 13 to 23 comprising the step of:

using reverse mapping definition information for defining a data mapping.

25. A computer program comprising program code means adapted to perform the steps of any of claims 13 to 24, when said program is run on a computer.

26. An intermediary system comprising the apparatus of any of claims 1 to 12.

ABSTRACT

APPARATUS, METHOD AND COMPUTER PROGRAM FOR DEFINING A DATA MAPPING BETWEEN TWO OR MORE DATA STRUCTURES

The invention relates to an apparatus for defining a data mapping between two or more data structures comprising: two or more data structures comprising incompatible identifiers; storage for storing said two or more data structures; means for selecting said two or more data structures; and means for deriving a definition of a data mapping between data elements represented by said incompatible identifiers, wherein said means for deriving a data mapping definition is operable to analyse previous data mapping definition information.

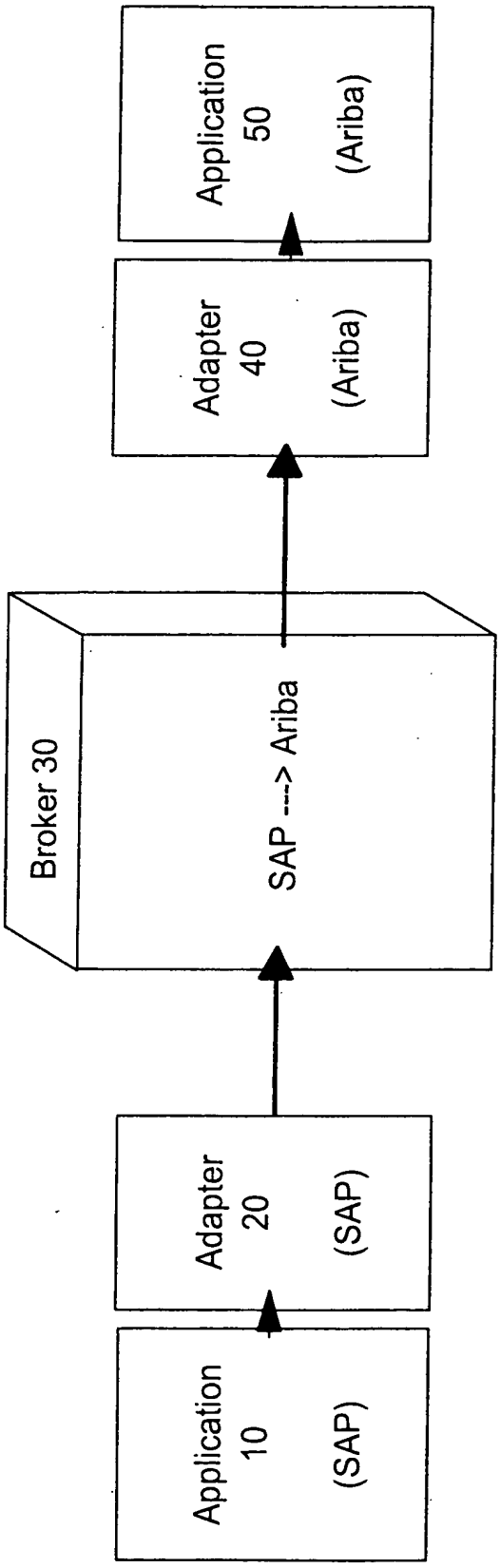

Figure 1a

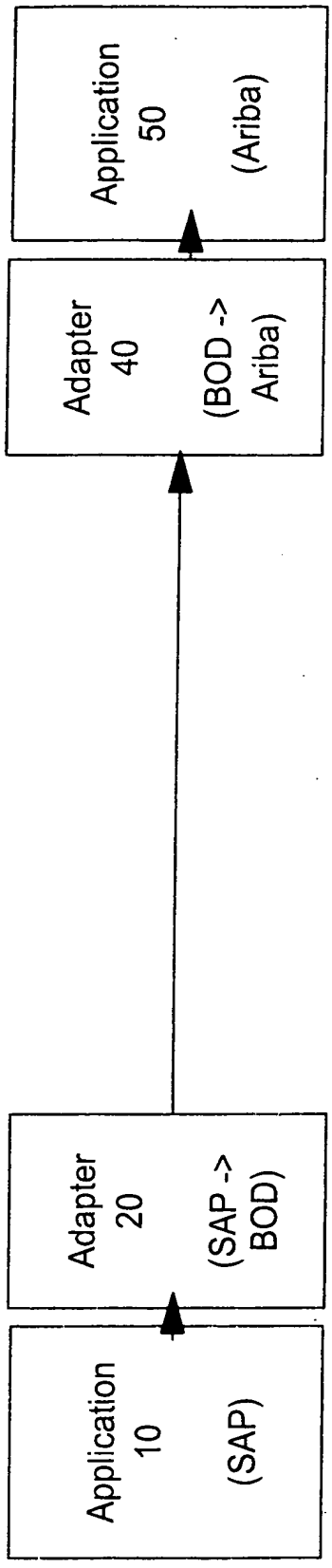

Figure 1b

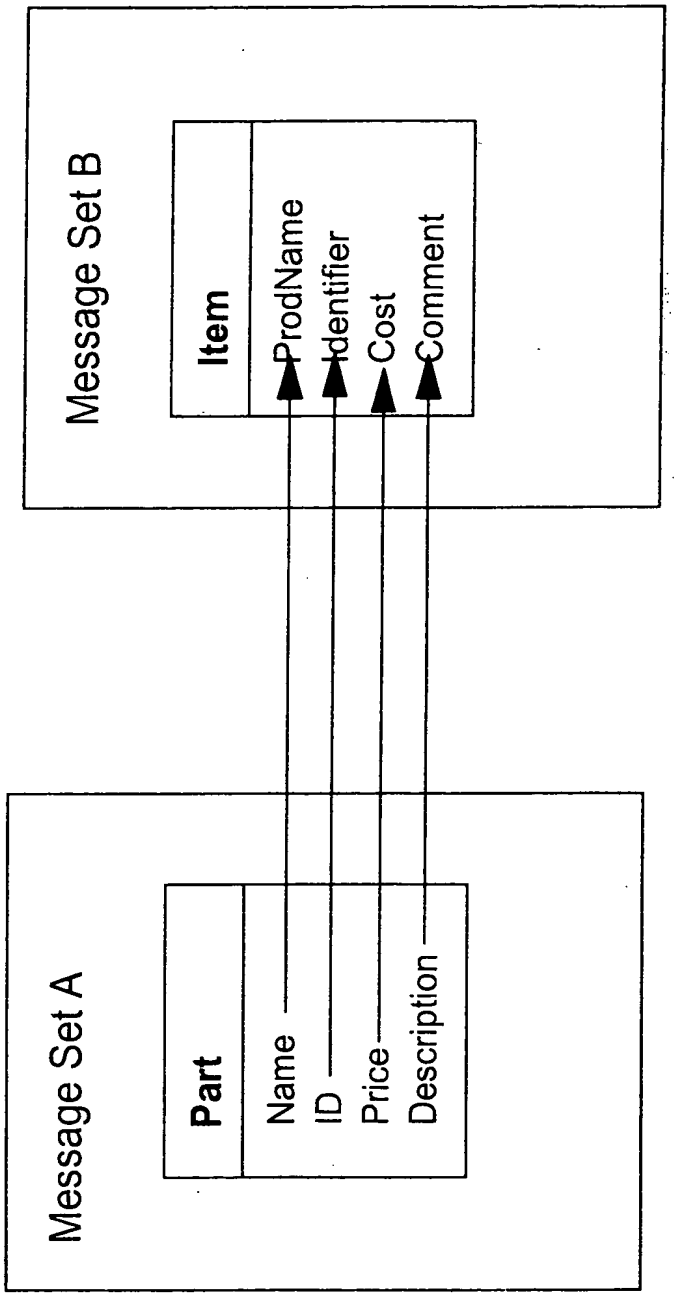
Message Set B
Item
ProdName
Identifier
Cost
Comment
Message Set A
Part
Name
ID
Price
Description

Figure 2

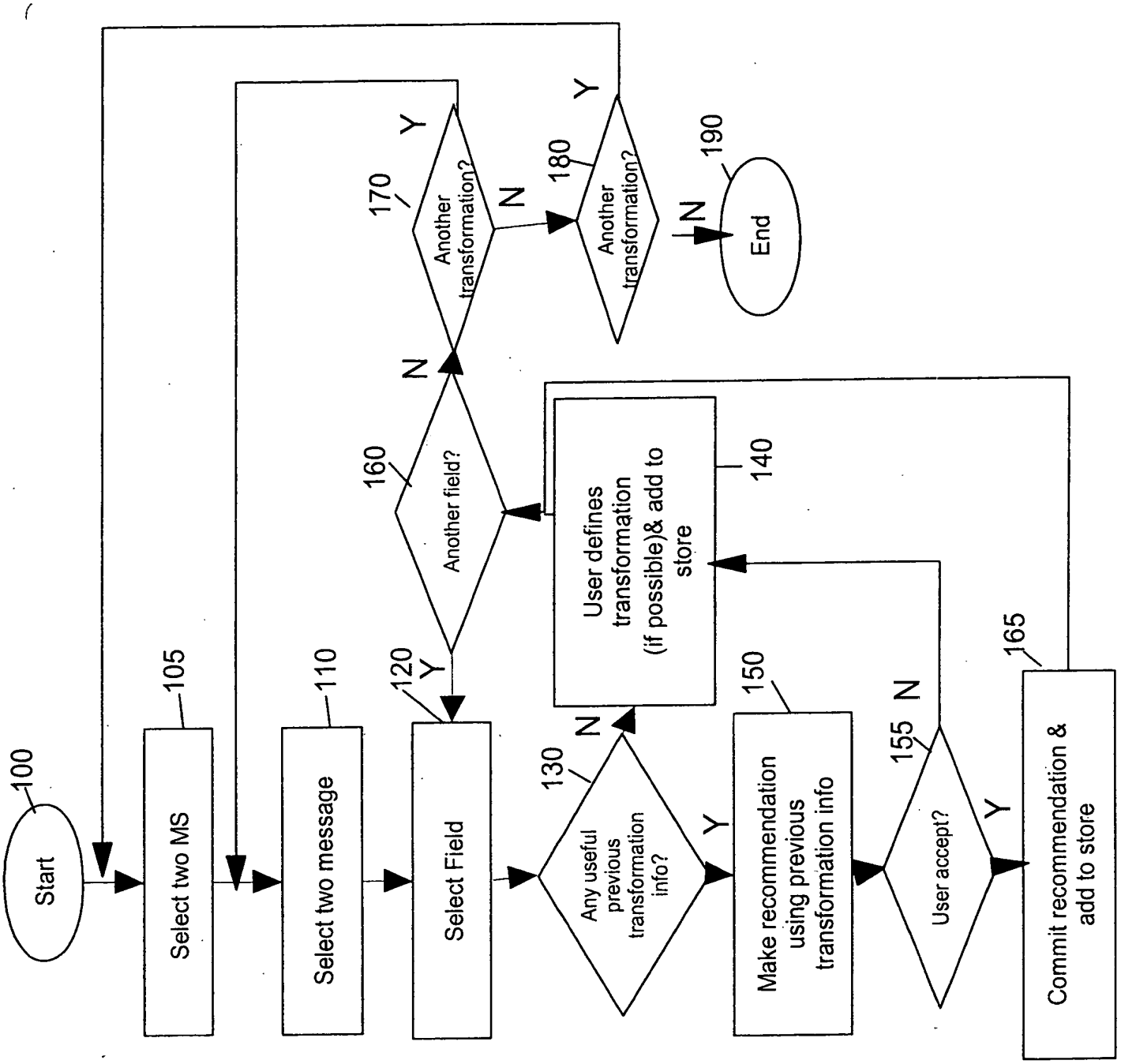
Start
100
Select two MS
105
Select two message
110
Select Field
120
Y
Any useful previous transformation info?
130
N
Y
Make recommendation using previous transformation info
150
User accept?
155
N
Y
Commit recommendation & add to store
165
User defines transformation (if possible)& add to store
140
Another field?
160
N
Y
Another transformation?
170
Y
N
Another transformation?
180
Y
N
End
190

Figure 3a

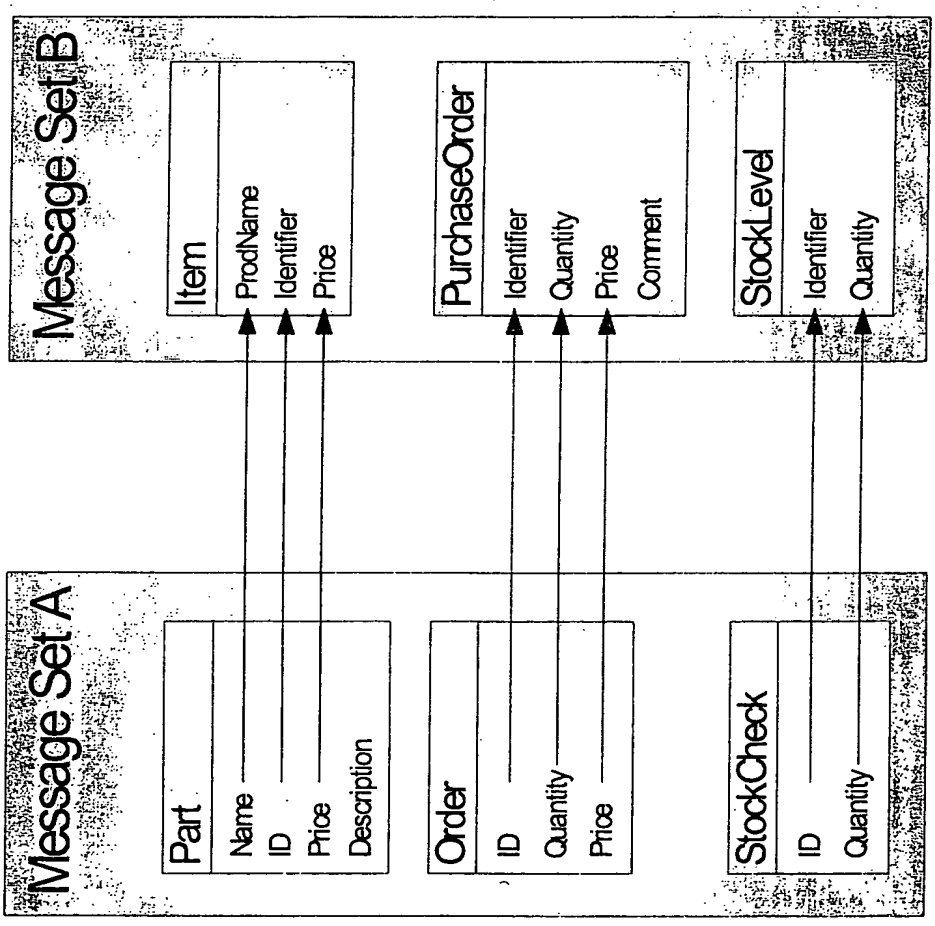
Message Set A
Part
Name
ID
Price
Description
Order
ID
Quantity
Price
StockCheck
ID
Quantity
Message Set B
Item
ProdName
Identifier
Price
PurchaseOrder
Identifier
Quantity
Price
Comment
StockLevel
Identifier
Quantity

Figure 3b

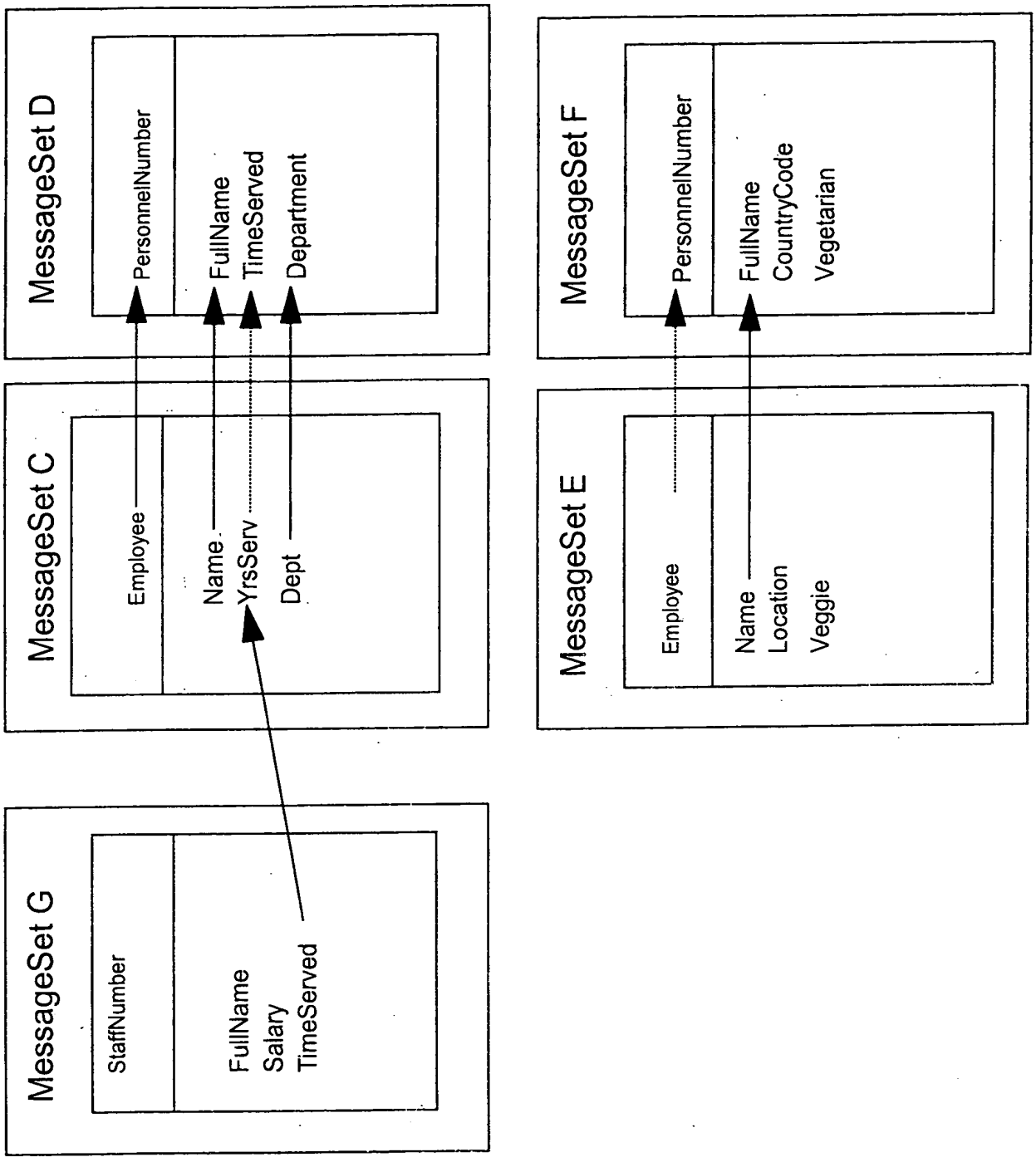
MessageSet D
PersonnelNumber
FullName
TimeServed
Department
MessageSet C
Employee
Name
YrsServ
Dept
MessageSet G
StaffNumber
FullName
Salary
TimeServed
MessageSet F
PersonnelNumber
FullName
CountryCode
Vegetarian
MessageSet E
Employee
Name
Location
Veggie

Figure 4

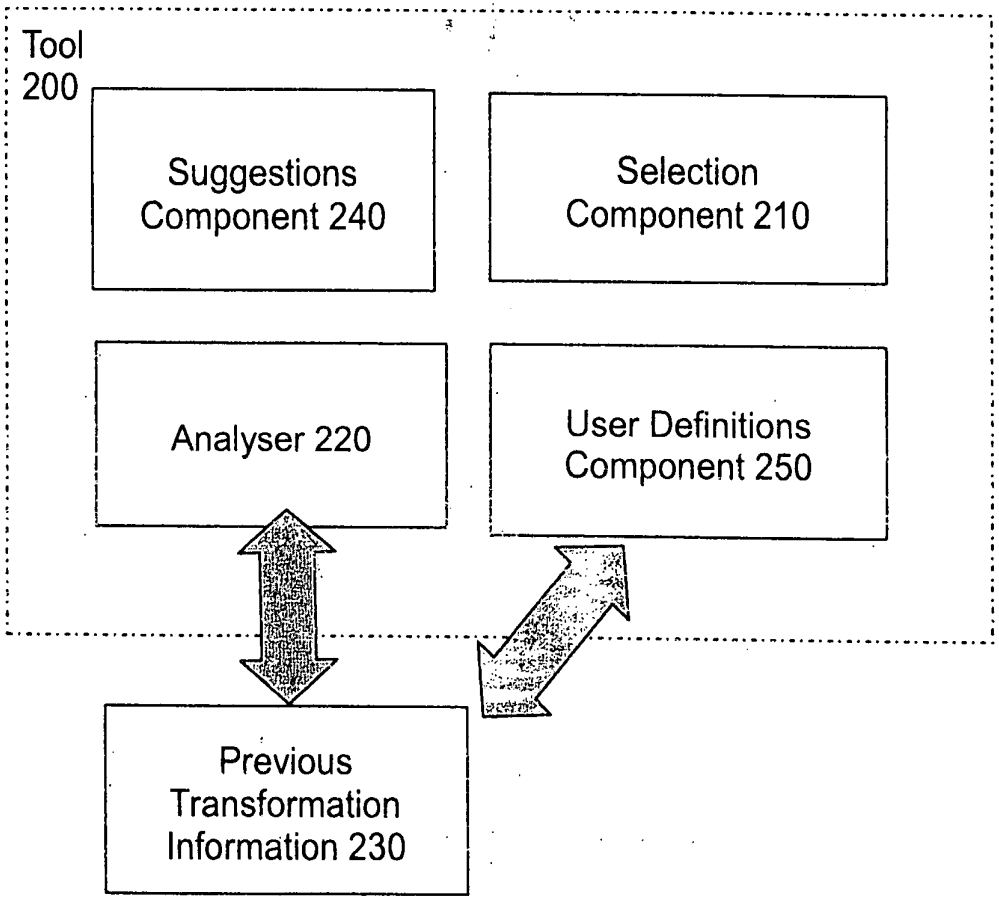
Tool
200
Suggestions
Component 240
Selection
Component 210
Analyser 220
User Definitions
Component 250
Previous
Transformation
Information 230

Figure 5